REMARKS

Applicant amended independent claim 1 to clarify that a plurality of memory access partitions are generated, and to recite a feature, similar to that previously recited in claim 5, namely, that the plurality of partitions contain corresponding subsets of the intermediate standard format instructions, and that the plurality of memory access partitions is directed to specific memory banks. Support for the amended language is provided, for example, at page 4, lines 1-11 of the originally filed application. Applicant similarly amended independent claims 15 and 27.

Additionally, applicant amended claims 4, 6, 16 and 30 to make the language recited therein consistent with the language of corresponding amended independent claims. Applicant also cancelled claim 5, and amended claims 6-12 to correct their dependency. After these amendments, claims 1-4 and 6-30 are pending in the above-identified application. Claims 1, 15 and 27 are independent.

The examiner rejected claims 1-6, 13-18 and 25-30 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,571,319 to Tremblay et al. The examiner further rejected claims 7-12 and 19-24 under 35 U.S.C. 103(a) as being unpatentable over Tremblay in view of the Microsoft Press Computer Dictionary.

Applicant's independent claim 1, is allowable over these references since no combination describes or suggests: "generating a plurality of memory access partitions containing corresponding subsets of the intermediary standard formatted memory access instructions, the plurality of the memory access partitions directed to specific memory banks."

Applicant's method, as recited in claim 1, converts the source-code memory access instructions into intermediary instructions, and generates partitions into which the intermediary instructions are grouped. The partitions are directed to specific memory banks of the processor-based device on which the program code to be generated will execute. Grouping the intermediate standard format instructions into groups directed to specific memory banks enables to identify candidate memory access instructions that can be combined into vector instruction because such candidate instructions access the same memory bank.

Tremblay describes methods and apparatus for combining data from a plurality of memory access transactions, and writing the combined data to memory in a single memory access transaction (col. 1, lines 10-15). Tremblay's apparatus includes instruction combining logic 200 that "comprises logic for processing store pair instructions before execution and determines whether and how to combine store pair instructions to form fewer memory store instructions" (col. 4, lines 39-42). Particularly, in relation to the functions performed by the instruction combining logic 200, Tremblay explains:

> **FIG. 3 shows a flow diagram representing the operation of instruction combining logic 200, in accordance with one aspect of the invention. The operation begins at step 300, where control logic 210 waits for a memory access transaction or instruction. When control logic 210 receives an instruction, instruction identification logic 220 determines whether the received instruction is a "store pair" instruction (step 305). If not, control logic 210 outputs the received instruction for execution (step 310). If instruction identification logic 220 determines that the received instruction is a "store pair" instruction, buffer logic 240 holds the received instruction in buffer 260 (step 315). In one embodiment, buffer logic 240 stores the data to be written and the address at which the data is to be written in buffer 260.**
>
> **Control logic 210 waits for the next instruction and begins programmable timer 250 (step 320). In a preferred embodiment, programmable timer 250 has previously been programmed to output a timeout signal if a preset amount of time has elapsed before being reset by control logic 210. If programmable timer 250 outputs a timeout signal before being reset, combined instruction generation logic 270 generates a combined instruction from the data and addresses held in buffer 260 and outputs the combined instruction for execution (step 325). If control logic 210 receives an instruction before programmable timer 250 outputs a timeout signal, operation flow moves to step 330.**
>
> **Instruction identification logic 220 determines whether the received instruction is a "store pair" instruction (step 330). If not, the received instruction is output for execution (step 335) and combined instruction generation logic 270 generates a combined instruction from data and addresses in buffer 260 and outputs the combined instruction for execution (step 325). If instruction identification logic 220 determines that the received instruction is a "store pair" instruction, operation flow moves to step 340.**
>
> **Cache line detection logic 230 determines whether the received "store pair" instruction includes data from the same cache line as the data stored in buffer 260 (step 340). If not, combined instruction generation logic 270 generates a combined instruction from the data and addresses stored in buffer 260 and outputs the combined instruction for execution (step 345). Buffer logic 240 then stores the data and address for the received instruction in buffer 260 and operation flow returns to step 320. If cache**

AN line detection logic 230 determines that the received "store pair" instruction includes data from the same cache line as the data stored in buffer 260, buffer logic 240 stores the data and address for the received instruction in buffer 260 (step 360) and operation flow moves to step 365. (emphasis added, col. 5, line 31, to col. 6, line 12)

Tremblay's apparatus maintains at any given time only a single sequence of candidate memory access instructions that may be combined. After a particular instruction has been identified as a "store pair instruction" (i.e., an instruction that causes data to be written to the main memory) and stored in the apparatus' buffer 260, Tremblay's apparatus retrieves the next instruction. If the retrieved instruction is neither a "store pair" instruction, nor a "store pair" instruction holding data from the same cache line as the data corresponding to the "store pair" instructions already in the buffer 260, Tremblay's apparatus causes the "store pair" instructions stored in the buffer 260 to be combined to a single instruction and be immediately executed. The just retrieved instruction is then stored in the vacated buffer 260 if it is a "store pair instruction", or it is executed if it is not a "store pair instruction". Tremblay, therefore, does not group memory access instructions into a plurality of partitions, or sequences, that correspond to specific memory banks. Accordingly, Tremblay neither discloses nor suggests at least the feature of "generating a plurality of memory access partitions containing corresponding subsets of the intermediary standard formatted memory access instructions, the plurality of memory access partitions directed to specific memory banks," as required by applicant's independent claim 1.

Additionally, independent claim 1 also recites "generating a match set of instruction patterns including matches of instruction patterns to the corresponding subsets of the intermediary standard formatted memory access instructions in the plurality of memory access partitions." In contrast, Tremblay determines if a particular retrieved instruction is a "store pair instruction" prior to placing that instruction in the buffer 260 (if in fact that retrieved instruction is a "store pair instruction".) As Tremblay further describes, no additional processing is performed on any of the instructions placed in buffer 260 other than outputting the instructions in the buffer as a combined instruction (see col. 5, line 31, to col. 6, line 12.) In other words, Tremblay neither describes nor suggests "generating a match set of instructions patterns including matches of instruction patterns to the corresponding subsets of the intermediary

standard formatted memory access instructions in the plurality of memory access partitions," as required by applicant's independent claim 1.

Therefore, applicant's independent claim 1 is patentable over the cited art.

Claims 2-4 and 6-14 depend from independent claim 1 and are therefore patentable for at least the same reasons as independent claim 1.

Independent claim 15 recites "grouping subsets of the intermediary memory access instructions into a plurality of memory access partitions, with the plurality of memory access partitions containing intermediate memory access instructions directed to specific memory banks; and vectorizing the intermediary memory access instructions in the subsets corresponding to the plurality of memory access partitions that match instruction patterns." For reasons similar to those provided with respect to independent claim 1, at least these features are not disclosed by Tremblay. Applicant's independent claim 15 is therefore patentable over the cited art.

Claims 16-26 depend from independent claim 15 and are therefore patentable for at least the same reasons as independent claim 15.

Independent claim 27 recites "generate a plurality of memory access partitions containing corresponding subsets of the intermediary standard formatted memory access instructions, with the plurality of memory access partitions directed to specific memory banks; generate a match set of instruction patterns including matches of instruction patterns to the corresponding subsets of the intermediary standard formatted memory access instructions in the plurality of memory access partitions." For reasons similar to those provided with respect to independent claim 1, at least these features are not disclosed by the cited art. Accordingly, independent claim 27 is patentable over the cited art.

Claims 28-30 depend from independent claim 27 and are therefore patentable for at least the same reasons as independent claim 27.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

In view of the foregoing, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

Please apply any required fees to deposit account 06-1050, referencing the attorney docket number shown above.

Respectfully submitted,

Date: December 22, 2006

Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906